

BUNDESREPUBLIK DEUTSCHLAND

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)



PCT EP 03/08081

REC'D 14 NOV 2003

WIPO

PCT

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen:

103 15 295.4

Anmeldetag:

04. April 2003

Anmelder/Inhaber:

PACT XPP Technologies AG, München/DE

Bezeichnung:

Verfahren und Vorrichtung für die
Datenverarbeitung

IPC:

G 06 F 12/08

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 5. September 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

CERTIFIED COPY OF
PRIORITY DOCUMENT

Stanschus



Deutsche Patentanmeldung

Anmelder: PACT XPP Technologies AG
Muthmannstrasse 1
D-80939 München

Vertreter: Patentanwalt
Claus Peter Pietruk
Heinrich-Lilienfein-Weg 5
D-76229 Karlsruhe
Vertreter-Nr. 321 605

Titel: Verfahren und Vorrichtung
für die Datenverarbeitung

Beschreibung

Die vorliegende Erfindung betrifft das oberbegrifflich Beanspruchte und befasst sich somit mit Verbesserungen bei der Verwendung von rekonfigurierbaren Prozessortechnologien für die Datenverarbeitung.

Verwiesen wird bezüglich des bevorzugten Aufbaus von Logikzellenfeldern auf die XPP-Architektur und vorveröffentlichte sowie jüngere Schutzrechtsanmeldungen des vorliegenden Anmelders, die zu Offenbarungszwecken vollumfänglich eingliedert sind. Erwähnt seien somit insbesondere die
DE 44 16 881 A1, DE 197 81 412 A1, DE 197 81 483 A1,
DE 196 54 846 A1, DE 196 54 593 A1, DE 197 04 044.6 A1,
DE 198 80 129 A1, DE 198 61 088 A1, DE 199 80 312 A1,
PCT/DE 00/01869, DE 100 36 627 A1, DE 100 28 397 A1,

DE 101 10 530 A1, DE 101 11 014 A1, PCT/EP 00/10516,
EP 01 102 674 A1, DE 198 80 128 A1, DE 101 39 170 A1,
DE 198 09 640 A1, DE 199 26 538.0 A1, DE 100 50 442 A1, sowie
die PCT/EP 02/02398, DE 102 40 000, DE 102 02 044,
5 DE 102 02 175, DE 101 29 237, DE 101 42 904, DE 101 35 210,
EP 01 129 923, PCT/EP 02/10084, DE 102 12 622, DE 102 36 271,
DE 102 12 621, EP 02 009 868, DE 102 36 272, DE 102 41 812,
DE 102 36 269, DE 102 43 322, EP 02 022 692, ebenso wie die
EP 02 001 331 und die EP 02 027 277.

10

Ein Problem bei herkömmlichen Ansätzen zu rekonfigurierbaren
Technologien besteht dann, wenn die Datenverarbeitung primär
auf einer sequenziellen CPU unter Hinzuziehung eines konfigu-
rierbaren Datenverarbeitungslogikzellenfeldes oder derglei-
15 chen erfolgen soll und/oder eine Datenverarbeitung gewünscht
ist, in der viele und/oder umfangreiche sequenziell auszufüh-
rende Verarbeitungsschritte vorliegen.

20

Es sind Ansätze bekannt, die sich damit befassen, wie eine
Datenverarbeitung sowohl auf einem konfigurierbaren Datenver-
arbeitungslogikzellenfeld als auch auf einer CPU erfolgen
kann.

25

So ist aus der WO 00/49496 ein Verfahren zum Ausführen eines
Computerprogrammes mit einem Prozessor bekannt, der eine kon-
figurierbare funktionelle Einheit umfasst, die in der Lage
ist, rekonfigurierbare Anweisungen auszuführen, deren Effekt
zur Laufzeit durch Laden eines Konfigurationsprogrammes rede-
finiert werden kann, wobei das Verfahren die Schritte um-
30 fasst, daß Kombinationen rekonfigurierbarer Anweisungen aus-
gewählt, ein respektives Konfigurationsprogramm für jede Kom-
bination erzeugt und das Computerprogramm ausgeführt wird.

CONFIDENTIAL

Dabei soll jedes Mal, wenn eine Anweisung aus einer der Kombinationen während der Ausführung gebraucht wird und die konfigurierbare funktionelle Einheit nicht mit dem Konfigurationsprogramm für diese Kombination konfiguriert ist, das Konfigurationsprogramm für alle der Anweisungen der Kombination in die konfigurierbare funktionelle Einheit geladen werden. Weiter ist aus der WO 02/50665 A1 eine Datenverarbeitungsvorrichtung mit einer konfigurierbaren funktionellen Einheit bekannt, wobei die konfigurierbare funktionelle Einheit dazu dient, eine Anweisung gemäß einer konfigurierbaren Funktion auszuführen. Die konfigurierbare funktionelle Einheit weist eine Vielzahl von unabhängigen konfigurierbaren Logikblöcken zum Ausführen programmierbarer Logikoperationen auf, um die konfigurierbare Funktion zu implementieren. Konfigurierbare Verbindungsschaltkreise sind zwischen den konfigurierbaren Logikblöcken und sowohl den Eingängen als auch den Ausgängen der konfigurierbaren funktionellen Einheit vorgesehen. Dies erlaubt eine Optimalisierung der Verteilung von Logikfunktionen über die konfigurierbaren Logikblöcke.

Ein Problem bei herkömmlichen Architekturen besteht dann, wenn eine Ankopplung erfolgen soll und/oder Technologien wie Datastreaming, Hyperthreading, Multithreading und so weiter in sinnvoller und Performance steigernder Weise ausgenutzt werden sollen. Die beispielhaft erwähnte Technologie der vorzitierten Nicht-Anmelder-Dokumente zeigt etwa eine Anordnung, bei der zwar Konfigurationen in ein konfigurierbares Datenverarbeitungslogikzellenfeld geladen werden können, bei welchen allerdings der Datenaustausch zwischen der ALU der CPU und dem konfigurierbaren Datenverarbeitungslogikzellenfeld, sei es ein FPGA, DSP oder dergleichen, über die Register erfolgt. Mit anderen Worten müssen Daten aus einem Datenstrom

CONFIDENTIAL

zunächst sequenziell in Register geschrieben werden und dann sequenziell wieder in diesen abgelegt werden. Auch ist ein Problem dann gegeben, wenn ein Zugriff auf Daten von extern erfolgen soll, da selbst dann noch Probleme beim zeitlichen

5 Ablauf der Datenverarbeitung im Vergleich zur ALU und bei der Zuweisung von Konfigurationen und so weiter bestehen. Die herkömmlichen Anordnungen, wie sie aus den Nicht-Anmelder-eigenen Schutzrechten bekannt sind, werden unter anderem dazu verwendet, Funktionen im konfigurierbaren Datenverarbeitungs-
10 logikzellenfeld, DFP, FPGA oder dergleichen abzuarbeiten, die nicht effizient auf der CPU-eigenen ALU abzuarbeiten sind.

Damit wird das konfigurierbare Datenverarbeitungslogikzellenfeld praktisch verwendet, um benutzerdefinierte Opcodes zu ermöglichen, die eine effizientere Abarbeitung von Algorithmen ermöglichen, als dies auf dem ALU-Rechenwerk der CPU ohne
15 konfigurierbare Datenverarbeitungslogikzellenfeldunterstützung möglich wäre.

Im Stand der Technik ist, wie erkannt wurde, die Ankopplung demnach im Regelfall wortbasiert, nicht jedoch blockbasiert,
20 wie es zur datenströmenden Verarbeitung erforderlich wäre. Es ist zunächst wünschenswert, eine effizientere Datenverarbeitung zu ermöglichen, als dies mit einer engen Ankopplung über Register der Fall ist.

25 Eine weitere Möglichkeit zur Verwendung von Logikzellenfeldern aus grob- und/oder feingranular gebäuten Logikzellen und Logikzellenelementen besteht in einer sehr losen Ankopplung eines solchen Feldes an eine herkömmliche CPU und/oder ein
30 CPU-Kern bei eingebetteten Systemen. Hierbei kann ein herkömmliches, sequenzielles Programm auf einer CPU oder dergleichen laufen, beispielsweise ein in C, C++ oder derglei-



CONFIDENTIAL

felder nach Prinzipien der Modelle für sequenziell arbeitende Maschinen ist weiterhin schwierig.

Aus dem Stand der Technik ist weiter bekannt, dass mehrere Konfigurationen, die eine jeweils unterschiedliche Funktionsweise von Arrayteilen bewirken, simultan auf dem Prozessorfeld (PA) abgearbeitet werden können und dass ein Wechsel von einer oder einigen der Konfiguration(en) ohne Störung anderer zur Laufzeit erfolgen kann. Es sind Verfahren und in Hardware implementierte Mittel zu deren Umsetzung bekannt, wie sichergestellt werden kann, dass dabei ein Abarbeiten von auf das Feld zu ladenden Teilkonfigurationen ohne Deadlock erfolgen kann. Verwiesen wird hierzu insbesondere auf die die Filmotechnik betreffenden Anmeldungen PACT05 (DE 196 54 593.5-53, WO 98/31102) PACT10 (DE 198 07 872.2, WO 99/44147, WO 99/44120) PACT13 (DE 199 26 538.0, WO 00/77652), PACT17 (DE 100 28 397.7, WO 02/13000). Diese Technologie ermöglicht in gewisser Weise bereits eine Parallelisierung und, bei entsprechender Gestaltung und Zuordnung der Konfigurationen, auch eine Art Multitasking/Multithreading und zwar dergestalt, dass eine Planung, das heißt ein Scheduling und/oder eine Zeitnutzungsplanungssteuerung vorgesehen ist. Es sind also aus dem Stand der Technik schon Zeitnutzungsplanungssteuerungsmittel und -verfahren per se bekannt, die, zumindest unter entsprechender Zuordnung von Konfigurationen zu einzelnen Aufgaben und/oder Fäden zu Konfigurationen und/oder Konfigurationsfolgen, ein Multitasking und/oder Multithreading erlauben. Die Verwendung solcher Zeitnutzungsplanungssteuermittel, die im Stand der Technik zur Konfigurierung und/oder Konfigurationsverwaltung verwendet wurden, zu Zwecken des Scheduling von Tasks, Threads, Multi- und Hyperthreads wird per se als erfinderisch angesehen.

5

Der Grundgedanke der Erfindung besteht darin, Neues für die gewerbliche Anwendung bereitzustellen.

Die Lösung dieser Aufgabe wird in unabhängiger Form beansprucht. Bevorzugte Ausführungsformen finden sich in den Unteransprüchen.

15 Ein erster wesentlicher Aspekt der vorliegenden Erfindung ist somit darin zu sehen, dass dem Datenverarbeitungslogikzellenfeld Daten im Ansprechen auf die Ausführung einer Ladekonfiguration durch das Datenverarbeitungslogikzellenfeld zugeführt werden und/oder Daten aus diesem Datenverarbeitungslogikzellenfeld weggeschrieben (STORE) werden, indem eine
20 STORE-Konfiguration entsprechend abgearbeitet wird. Diese Lade- und oder Speicherkonfigurationen sind dabei bevorzugt derart auszugestalten, dass innerhalb des Datenverarbeitungslogikzellenfeldes direkt oder indirekt Adressen jener Speicherstellen generiert werden, auf welche ladend und/oder
25 speichernd direkt oder indirekt zugegriffen werden soll. Es ist durch diese Einkonfiguration von Adressgeneratoren innerhalb einer Konfiguration möglich, eine Vielzahl von Daten in das Datenverarbeitungslogikzellenfeld einzuladen, wo sie gegebenenfalls in internen Speichern (iRAM) ablegbar sind
30 und/oder wo sie in internen Zellen wie EALUs mit Registern und/oder dergleichen eigenen Speichermitteln abgelegt werden

CONFIDENTIAL

können. Die Lade- beziehungsweise Speicherkonfiguration ermöglicht somit ein blockweises und nahezu datenstromartiges, insbesondere gegenüber Einzelzugriff vergleichsweises schnelles Laden von Daten und es kann eine solche Lade-Konfiguration ausgeführt werden vor einer oder mehreren tatsächlich Daten auswertend und/oder verändernd abarbeitenden Konfiguration(en), mit welcher/n die vorab geladenen Daten verarbeitet werden. Das Datenladen kann dabei typisch bei großen Logikzellenfeldern in kleinen Teilbereichen derselben geschehen, während andere Teilbereiche mit anderen Aufgaben befaßt sind. Bei der in anderen veröffentlichten Dokumenten des Anmelders beschriebenen Ping-Pong-artigen Datenverarbeitung, bei der auf beiden Seiten eines Datenverarbeitungsfeldes Speicherzellen vorgesehen sind, wobei die Daten in einem ersten Verarbeitungsschritt von dem Speicher auf der einen Seite durch das Datenverarbeitungsfeld zum Speicher auf der anderen Seite strömen, dort die beim ersten Felddurchströmen erhaltenen Zwischenergebnisse im zweiten Speicher abgelegt werden, gegebenenfalls das Feld umkonfiguriert wird, die Zwischenergebnisse dann für die Weiterverarbeitung zurückströmen usw., kann etwa eine Speicherseite durch eine LOAD-Konfiguration in einem Array-Teil mit neuen Daten vorgeladen werden, während aus der gegenüberliegenden Speicherseite Daten mit einer STORE-Konfiguration in einem anderen Array-Teil weggeschrieben werden. Dieses simultane LOAD/STORE-Vorgehen ist im übrigen auch ohne räumliche Speicherbereichstrennung möglich.

Das Laden kann insbesondere aus einem Cache und in diesen hinein erfolgen. Dies hat die Vorteile, dass die externe Kommunikation mit größeren Speicherbänken über den Cachecontroller gehandhabt wird, ohne dass innerhalb des Datenverarbeitungslogikzellenfeldes separate Schaltanordnungen dafür vor-

CONFIDENTIAL

gesehen sein müssen, dass der Zugriff in lesender oder schreibender Weise bei Cache-Speichermitteln typisch sehr schnell und mit allenfalls geringer Latenzzeit erfolgen wird und dass auch typisch eine CPU-Einheit, dort typisch über eine separate LOAD/STORE-Einheit, an diesen Cache angebunden ist, sodass ein Zugriff auf Daten und ein Austausch derselben zwischen CPU-Kern und Datenverarbeitungslogikzellenfeld blockweise schnell und derart erfolgen kann, dass nicht für jedes Übergeben von Daten ein separater Befehl etwa aus dem OpCode-Fetcher der CPU abgeholt und verarbeitet werden muss.

Es erweist sich diese Cacheankoppelung auch als wesentlich günstiger als eine Ankopplung eines Datenverarbeitungslogikzellenfeldes an die ALU über Register, wenn diese Register nur über eine LOAD/STORE-Einheit mit einem Cache kommunizieren, wie dies aus den Nicht-PACT-eigenen zitierten Schriften per se bekannt ist.

Es kann eine weitere Datenverbindung zu der Lade/Speicher-einheit der oder einer dem Datenverarbeitungslogikzellenfeld zugeordneten Sequenziell-CPU-Einheit vorgesehen sein und/oder zu deren Register.

Es sei erwähnt, dass ein Ansprechen derartiger Einheiten über separate Eingangs-/Ausgangsanschlüsse (IO-Ports) der insbesondere als VPU beziehungsweise XPP ausgestaltbaren Datenverarbeitungslogikzellenanordnung erfolgen kann und/oder durch einen oder mehrere einem Einzelport nachgeschaltete Multiplexer.

Dass neben dem insbesondere blockweisen und/oder streamenden und/oder im Random-Access, insbesondere im RMW-Modus (Read-

vermag, sofern entsprechende Anwendung benötigt werden. Es sei darauf hingewiesen, dass vorstehend von FPGA-artigen vorbeziehungsweise nachgeschalteten Strukturen die Rede war. Mit FPGA-artig muss aber, was explizit erwähnt sei, nicht zwingend Bezug genommen sein auf 1-Bit-granulare Anordnungen. Es ist insbesondere möglich, statt dieser hyperfeingranularen Strukturen lediglich feiner granulare Strukturen von zum Beispiel 4 Bit Breite vorzusehen. Das heißt, die FPGA-artigen Eingangs- und/oder Ausgangsstrukturen vor und/oder nach einer insbesondere als SIMD-Rechenwerk ausgestalteten ALU-Einheit sind so konfigurierbar, dass immer 4 Bit breite Datenwörter zugeführt und/oder verarbeitet werden. Es ist möglich, hier eine Kaskadierung vorzusehen, so dass zum Beispiel die einkommenden 32 Bit breiten Datenwörter in 4 separierte bzw. separierende 8-Bit-FPGA-artige, nebeneinander angeordnete Strukturen strömen, diesen 4 Stück 8 Bit breiten FPGA-artigen Strukturen ein zweiter Streifen mit 8 Stück 4 Bit breiten FPGA-artigen Strukturen nachgesetzt ist, und gegebenenfalls nach einem weiteren derartigen Streifen dann, sofern dies für den jeweiligen Zweck als erforderlich erachtet wird, zum Beispiel 16 Stück parallel nebeneinander angeordnete 2 Bit breite FPGA-artige Strukturen vorgesehen werden. Wenn dies der Fall ist, kann gegenüber rein hyperfeingranular FPGA-artigen Strukturen eine beträchtliche Verringerung des Konfigurationsaufwandes erzielt werden. Dass dies überdies dazu führt, dass der Konfigurationsspeicher und so weiter der FPGA-artigen Struktur wesentlich kleiner ausfallen kann und somit eine Einsparung an Chipfläche erzielt wird, sei erwähnt.

Prinzipiell sind die vorstehend beschriebenen Kopplungsvorteile bei Datenblockströmen über den Cache prinzipiell erreichbar; besonders bevorzugt ist es jedoch, wenn der Cache



CONFIDENTIAL

11

tungsprogrammaufspaltungstechnologien wie Multitasking, Multithreading und Hyperthreading. Der sich ergebende Vorteil, dass ein Pfadbala-

5 von Pipelinestufen durchlaufen werden können, Taktungen in unterschiedlicher Weise mögliche sind und so weiter. Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass durch das Hineinkonfigurieren einer Ladekonfiguration beziehungsweise einer Storekonfiguration in das XPP oder andere
10 Datenverarbeitungslogikzellenfelder die Daten in das Feld mit einer Geschwindigkeit hineingeladen werden oder aus diesem herausgeschrieben werden können, die nicht mehr bestimmt ist durch die Taktgeschwindigkeit der CPU, die Geschwindigkeit, mit welcher der OpCode-Fetcher arbeitet, oder dergleichen.

15 Mit anderen Worten ist die Ablaufsteuerung der Sequenziell-CPU nicht mehr flaschenhalsartig begrenzend für den Datendurchsatz des Datenzellenlogikfeldes, ohne dass eine nur noch lose Ankopplung besteht.

20 Während es in einer besonders bevorzugten Variante der Erfindung möglich ist, die für eine XPP-Einheit bekannte CT (bzw. CM; Konfigurationsmanager bzw. Konfigurationstabelle) zu verwenden, um sowohl das Konfigurieren eines oder mehrerer, auch hierarchisch mit mehreren CTs angeordneter XPP-Felder und

25 gleichzeitig eines oder mehrerer Sequenziell-CPU's, dort quasi als Hyperthreading-Hardwareverwaltung zu verwenden, was den inhärenten Vorteil hat, daß bekannte Technologien wie FILMO usw. für die hardwareunterstützte Verwaltung beim Hyperthreading einsetzbar werden, ist es alternativ und/oder, insbesondere
30 in hierarchischer Anordnung, zusätzlich möglich, dass ein Datenverarbeitungslogikzellenfeld wie eine XPP Konfigurationen vom OpCode-Fetcher einer Sequenziell-CPU über das Ko-

X

prozessor-Interface erhält. Dies führt dazu, daß von der Sequenziell-CPU und/oder einer anderen XPP ein Aufruf instantiiert werden kann, der zu einer Datenabarbeitung auf der XPP führt. Die XPP wird dabei dann z. B. über die beschriebene
5 Cache-Ankopplung und/oder mittels LOAD- und/oder STORE-Konfigurationen, die Adressgeneratoren für Laden und/oder Wegschreiben von Daten im XPP- bzw. Datenverarbeitungslogikzellenfeld vorsehen, im Datenaustausch gehalten. Mit anderen Worten wird eine Koprozessor-artige Ankopplung eines Datenverarbeitungslogikzellenfeldes möglich, während gleichzeitig ein datenstromartiges Datenladen durch Cache- und/oder
10 I/O-Port-Kopplung erfolgt.

Es sei erwähnt, daß die Koprozessor-Ankopplung, d. h. die Ankopplung des Datenverarbeitungslogikzellenfeldes typisch dazu
15 führen wird, daß das Scheduling auch für dieses Logikzellenfeld auf der Sequenziell-CPU oder einer dieser übergeordneten Schedulereinheit bzw. einem entsprechenden Schedulermittel erfolgen wird. In einem solchen Fall findet praktisch die
20 Threading-Kontrolle und -verwaltung auf dem Scheduler bzw. der Sequenziell-CPU statt. Obwohl dies per se möglich ist, wird dies, zumindest bei einfachster Implementierung der Erfindung, nicht zwingend der Fall sein. Vielmehr kann eine Verwendung des Datenverarbeitungslogikzellenfeldes durch Aufruf in herkömmlicher Weise wie bei einem Standard-Koprozessor
25 etwa bei 8086/8087-Kombinationen erfolgen.

Weiter sei erwähnt, daß es in einer besonders bevorzugten Variante, unabhängig von der Art der Konfiguration, sei es über
30 das Koprozessor-Interface, den als Scheduler mitdienenden Konfigurationsmanager (CT) der XPP bzw. des Datenverarbeitungslogikzellenfeldes oder dergleichen oder auf andere Wei-

se, möglich ist, im bzw. unmittelbar am Datenverarbeitungslogikzellenfeld bzw. unter Verwaltung des Datenverarbeitungslogikzellenfeldes Speicher, insbesondere interne Speicher, insbesondere bei der XPP-Architektur, wie sie aus den diversen

5 Voranmeldungen und den Veröffentlichungen des Anmelders bekannt ist, RAM-PAEs, oder andere entsprechend verwaltete oder interne Speicher wie ein Vektorregister anzusprechen, d. h. die über die LOAD-Konfiguration eingeladenen Datenmengen vektorartig wie in Vektorregistern in die internen Speicher abzulegen, dann, nach Umkonfigurieren der XPP bzw. des Daten-

10 verarbeitungslogikzellenfeldes, also Überschreiben bzw. Nachladen und/oder Aktivieren einer neuen Konfiguration, die die eigentliche Verarbeitung der Daten durchführt (in diesem Zusammenhang sei darauf hingewiesen, daß für eine solche Verarbeitungskonfiguration auch Bezug genommen werden kann auf eine Mehrzahl von Konfigurationen, die z. B. im Wave-Modus und/oder sequenziell nacheinander abzuarbeiten sind) zuzugreifen wie bei einem Vektorregister und dann die dabei erhaltenen Ergebnisse und/oder Zwischenergebnisse wiederum in

15 die internen oder über die XPP wie interne Speicher verwalteten externen Speicher, um dort diese Ergebnisse abzulegen. Die so vektorregisterartig mit Verarbeitungsergebnissen beschriebenen Speichermittel unter XPP-Zugriff sind dann, nach Rekonfigurieren der Verarbeitungskonfiguration durch Laden

20 der STORE-Konfiguration in geeigneter Weise weggeschrieben, was wiederum datenstromartig geschieht, sei es über den I/O-Port direkt in externe Speicherbereiche und/oder, wie besonders bevorzugt, in Cache-Speicherbereiche, auf welche dann zu einem späteren Zeitpunkt die Sequenziell-CPU und/oder andere

25 Konfigurationen auf der zuvor die Daten erzeugt habenden XPP oder einer anderen entsprechenden Datenverarbeitungseinheit zugreifen können.

30

Eine besonders bevorzugte Variante besteht darin, zumindest für bestimmte Datenverarbeitungsergebnisse und/oder Zwischenergebnisse als Speicher- bzw. Vektorregistermittel, in welchem bzw. welches die erhaltenen Daten abzulegen sind, nicht einen internen Speicher zu benutzen, in welchen Daten über eine STORE-Konfiguration in den Cache- oder einen anderen Bereich, auf welchen die Sequenziell-CPU oder eine andere Datenverarbeitungseinheit zugreifen können, wegzuschreiben sind, sondern statt dessen unmittelbar die Ergebnisse wegzuschreiben in entsprechende, insbesondere zugriffsreservierte Cachebereiche, die insbesondere Slice-artig organisiert sein können. Dies kann gegebenenfalls den Nachteil einer größeren Latenz haben, insbesondere wenn die Wege zwischen der XPP- oder Datenverarbeitungslogikzellenfeldeinheit und dem Cache so lang sind, daß die Signallaufzeiten ins Gewicht fallen, führt aber dazu, daß gegebenenfalls keine weitere STORE-Konfiguration benötigt wird. Es sei im übrigen erwähnt, daß eine derartige Abspeicherung von Daten in Cache-Bereiche einerseits, wie vorstehend beschrieben, dadurch möglich ist, daß der Speicher, in welchen geschrieben wird, physikalisch nahe beim Cache-Controller liegt und als Cache ausgestaltet ist, dass aber alternativ und/oder zusätzlich auch die Möglichkeit besteht, einen Teil eines XPP-Speicherbereiches, XPP-internen Speichers oder dergleichen, insbesondere bei RAM über PAEs unter die Verwaltung eines oder, nacheinander mehrerer Cache-Speichercontroller zu stellen. Dies hat dann Vorteile, wenn die Latenz beim Abspeichern der Verarbeitungsergebnisse, welche innerhalb des Datenverarbeitungslogikzellenfeldes bestimmt werden, gering gehalten werden soll, während die Latenz beim Zugriff auf den dann nur noch als „Quasi-

Cache" dienenden Speicherbereich durch andere Einheiten nicht oder nicht signifikant ins Gewicht fällt.

Es sei im übrigen erwähnt, daß auch eine Ausgestaltung derart möglich ist, daß der Cache-Controller einer herkömmlichen Sequenziell-CPU einen Speicherbereich als Cache anspricht, der, ohne dem Datenaustausch mit dem Datenverarbeitungslogikzellenfeld zu dienen, auf und/oder bei diesem physikalisch liegt. Dies hat den Vorteil, daß dann, wenn Anwendungen auf dem Datenverarbeitungslogikzellenfeld laufen, die einen allenfalls geringen lokalen Speicherbedarf haben, und/oder wenn auch nur wenige weitere Konfigurationen bezogen auf die zur Verfügung stehenden Speichermengen benötigt werden, diese einer oder mehreren Sequenziell-CPU's als Cache zur Verfügung stehen können. Es sei erwähnt, daß dann der Cache-Controller für die Verwaltung eines Cache-Bereiches mit dynamischem Umfang, d. h. variierender Größe ausgebildet sein kann und wird. Eine dynamische Cache-Umfangsverwaltung bzw. Cache-Umfangsverwaltungsmittel für die dynamische Cache-Verwaltung wird typisch die Arbeitslast auf der Sequenziell-CPU und/oder dem Datenverarbeitungslogikzellenfeld berücksichtigen. Mit anderen Worten kann beispielsweise analysiert werden, wie viele NOPs in einer gegebenen Zeiteinheit auf der Sequenziell-CPU vorliegen und/oder wie viele Konfigurationen im XPP-Feld in dafür vorgesehenen Speicherbereichen vorabgelegt sein sollen, um eine schnelle Umkonfiguration, sei es im Wege einer Wellenrekongfiguration oder auf andere Weise. Die hiermit offenbarte dynamische Cachegröße ist dabei insbesondere bevorzugt laufzeitdynamisch, d. h. der Chacecontroller verwaltet jeweils eine aktuelle Cachegröße, die sich von Takt zu Takt oder Taktgruppe ändern kann. Es sei im übrigen darauf hingewiesen, daß die Zugriffsverwaltung eines XPP- bzw. Da-

tenverarbeitungslogikzellenfeldes mit Zugriff als interner Speicher wie bei einem Vektorregister und als Cache-artiger Speicher für den externen Zugriff was die Speicherzugriffe angeht bereits beschrieben wurde in der DE 196 54 595 und der
 5 PCT/DE 97/03013 (PACT03). Die genannten Schriften sind durch Bezugnahme zu Offenbarungszwecken hiermit vollumfänglich eingegliedert.

Vorstehend wurde auf Datenverarbeitungslogikzellenfelder Bezug
 10 zug genommen, die insbesondere zur Laufzeit rekonfigurierbar sind. Es wurde diskutiert, dass bei diesen eine Konfigurationsverwaltungseinheit (CT bzw. CN) vorgesehen werden kann. Aus den diversen, zu Offenbarungszwecken unter Bezug genommenen Schutzrechten des Anmelders sowie seinen weiteren Veröffentlichungen ist die Verwaltung von Konfigurationen per se
 15 bekannt. Es sei nun explizit darauf hingewiesen, dass derartige Einheiten und deren Wirkungsweise, mit der insbesondere unabhängig von Ankopplungen an Sequenziell-CPU's etc. aktuell noch nicht benötigte Konfigurationen vorladbar sind, auch
 20 sehr gut nutzbar sind, um im Multitaskingbetrieb und/oder bei Hyperthreading und/oder Multithreading einen Task- beziehungsweise einen Thread- und/oder Hyperthreadwechsel zu bewirken. Dazu kann ausgenutzt werden, dass während der Laufzeit eines Threads oder Tasks in die Konfigurationsspeicher
 25 bei einer einzelnen oder einer Gruppe von Zellen des Datenverarbeitungslogikzellenfeldes, also beispielsweise einer PAE eines PAE-Feldes (PA) auch Konfigurationen für unterschiedliche Aufgaben, das heißt Tasks oder Threads beziehungsweise Hyperthreads geladen werden können. Dies führt dann dazu,
 30 dass bei einer Blockade eines Tasks oder Threads, etwa wenn auf Daten gewartet werden muss, weil diese noch nicht verfügbar sind, sei es, da sie von einer anderen Einheit noch nicht

generiert oder empfangen wurden, beispielsweise auf Grund von Latenzen, sei es, weil eine Ressource derzeit noch durch einen anderen Zugriff blockiert ist, dann Konfigurationen für einen anderen Task oder Thread vorladbar und/oder vorgeladen sind und auf diese gewechselt werden kann, ohne dass der Zeitoverhead für einen Konfigurationswechsel bei der insbesondere schattengeladenen Konfiguration abgewartet werden muss. Während es prinzipiell möglich ist, diese Technik auch dann zu verwenden, wenn innerhalb eines Tasks die wahrscheinlichste Weiterführung vorhergesagt wird und eine Vorhersage nicht zutrifft (prediction miss), wird diese Art des Betriebs bei vorhersagefreiem Betrieb bevorzugt sein. Bei Verwendung mit einer rein sequentiellen CPU und/oder mehreren rein sequentiellen CPUs wird somit durch die Zuschaltung eines Konfigurationsmanagers eine Hyperthreadingverwaltungshardware realisiert. Verwiesen sei hinsichtlich dessen insbesondere auf PACT10 (DE 198 07 872.2, WO 99/44147, WO 99/44120). Dabei kann es als ausreichend erachtet werden, insbesondere dann, wenn nur für eine CPU und/oder einige wenige Sequenziell-CPU's eine Hyperthreadingverwaltung gewünscht ist, auf bestimmte, in den speziell unter Bezug genommenen Schutzrechten beschriebene Teilschaltungen wie den FILMO zu verzichten. Insbesondere wird damit die Verwendung der dort beschriebenen Konfigurationsmanager mit und/oder ohne FILMO für die Hyperthreadingverwaltung für eine und/oder mehrere rein sequenziell arbeitende CPUs mit oder ohne Ankopplung an eine XPP oder ein anderes Datenverarbeitungslogikzellenfeld offenbart und hiermit für sich beansprucht. Es wird hierin eine für sich erfinderische Besonderheit gesehen. Es sei im Übrigen erwähnt, dass eine Vielzahl von CPUs realisiert werden kann mit den bekannten Techniken, wie sie insbesondere aus PACT31 (DE 102 12 621.6-53, PCT/EP 02/10572) bekannt sind, bei welchen

innerhalb eines Arrays eine oder mehrere Sequenziell-CPUs aufgebaut werden unter Ausnutzung eines oder mehrerer Speicherbereiche insbesondere im Datenverarbeitungslogikzellenfeld für den Aufbau der sequenziellen CPU, insbesondere als

5 Befehls- und/oder Datenregister. Auch sei darauf verwiesen, dass bereits in früheren Anmeldungen wie PACT02, (DE 196 51 075.9-53, WO 98/26356), PACT04 (DE 196 54 846.2-53, WO 98/29952), PACT08, (DE 197 04 728.9, WO 98/35299) offenbart wurde, wie Sequenzer mit Ring- und/oder Wahlfrei-Zugriff-

10 Speichern aufgebaut werden können.

Es sei darauf hingewiesen, dass ein Task- beziehungsweise Thread- und/oder Hyperthreadwechsel unter Verwendung der bekannten CT-Technologie derart erfolgen kann und bevorzugt

15 auch erfolgen wird, dass einem per se bekannten, Software-implementierten Betriebssystem-Scheduler oder dergleichen von der CT Performance-Scheiben und/oder Zeitscheiben zugeordnet werden, während welchen bestimmt wird, von welchen Tasks oder Threads nachfolgend welche Teile per se, unterstellt, dass

20 Ressourcen frei sind, abzuarbeiten sind. Dazu sei ein Beispiel wie folgt gegeben: Zunächst soll für einen ersten Task eine Adressfolge generiert werden, gemäß welcher während der Ausführung einer LOAD-Konfiguration Daten aus einem Cache-Speicher, an dem ein Datenverarbeitungslogikzellenfeld in der

25 beschriebenen Weise angekoppelt ist, geladen werden sollen. Sobald diese Daten vorliegen, kann mit der Abarbeitung einer zweiten, der eigentlichen Datenverarbeitungs-konfiguration, begonnen werden. Auch diese kann vorgeladen werden, da sicher feststeht, dass diese Konfiguration, sofern keine Interrupts

30 oder dergleichen einen vollständigen Taskwechsel erzwingen, auszuführen ist. In herkömmlichen Prozessoren ist nun das Problem des sogenannten Cache-Miss bekannt, bei dem die Daten

zwar angefordert werden, aber nicht im Cache für den Ladezugriff bereit liegen. Tritt ein solcher Fall in einer Kopplung gemäß der vorliegenden Erfindung auf, kann bevorzugt auf einen anderen Thread, Hyperthread und/oder Task gewechselt werden, der insbesondere zuvor von dem insbesondere softwareimplementierten Betriebssystem-Scheduler und/oder einer anderen hard- und/oder softwareimplementierten, entsprechend wirkenden Einheit für eine nächstmögliche Ausführung bestimmt wurde und demgemäß bevorzugt vorab in einen der verfügbaren Konfigurationsspeicher des Datenverarbeitungslogikzellenfeldes insbesondere im Hintergrund während der Ausführung einer anderen Konfiguration, beispielsweise der LOAD-Konfiguration, welche das Laden jener Daten, auf die nun gewartet wird, bewirkt hat, geladen wurde. Das für die Vorabkonfiguration ungestört von der tatsächlichen Verschaltung der insbesondere grobgranular ausgebildeten Datenverarbeitungslogikzellen des Datenverarbeitungslogikzellenfeldes separate Konfigurationsleitungen von der konfigurierenden Einheit zu den jeweiligen Zellen direkt und/oder über geeignete Bussysteme geführt sein können wie per se im Stand der Technik bekannt, sei hier noch einmal explizit erwähnt, da diese Ausbildung hier besonders bevorzugt ist, um ein ungestörtes Vorabkonfigurieren ohne Störung einer anderen, gerade laufenden Konfiguration zu ermöglichen. Wenn dann die Konfiguration, auf welche während beziehungsweise auf Grund des Task-Thread- und/oder Hyperthreadwechsels gewechselt wurde, abgearbeitet wurde, und zwar, bei bevorzugten nicht teilbaren, ununterbrechbaren und somit quasi atomaren Konfigurationen bis zum Ende abgearbeitet wurde, wird teilweise eine weitere andere Konfiguration wie vorbestimmt durch die entsprechenden Scheduler, insbesondere den betriebssystemartigen Scheduler festgelegt, abgearbeitet und/oder jene Konfiguration, zu welcher zuvor die zugehörige

LOAD-Konfiguration ausgeführt wurde. Vor der Ausführung einer Verarbeitungskonfiguration, zu welcher zuvor eine LOAD-Konfiguration ausgeführt wurde, kann insbesondere abgetestet werden, ob mittlerweile die entsprechenden Daten in das Array eingeströmt sind, also die Latenzzeit, wie sie typisch auftritt, verstrichen ist und/oder die Daten tatsächlich vorliegen.

Mit anderen Worten werden dann Latenzzeiten, wenn sie auftreten, weil z. B. Konfigurationen noch nicht einkonfiguriert sind, Daten noch nicht geladen und/oder Daten noch nicht weggeschrieben wurden, überbrückt und/oder verdeckt, indem Threads, Hyperthreads und/oder Tasks ausgeführt werden, welche schon vorkonfiguriert sind und welche mit Daten arbeiten, die schon verfügbar sind beziehungsweise die an Ressourcen weggeschrieben werden können, die für das Wegschreiben bereits zur Verfügung stehen. Auf diese Weise werden Latenzzeiten weitgehend überdeckt und es wird, eine hinreichende Anzahl von per se auszuführenden Threads, Hyperthreads und/oder Tasks unterstellt, eine praktisch 100%-ige Ausnutzung des Datenverarbeitungslogikzellenfeldes erreicht.

Mit dem beschriebenen System bezüglich Datenstrom-Fähigkeit bei gleichzeitiger Ankopplung an eine Sequenziell-CPU und/oder bezüglich der Ankopplung eines XPP-Array beziehungsweise Datenverarbeitungslogikzellenfeldes und simultan einer Sequenziell-CPU an eine geeignete Schedulereinheit wie einen Konfigurationsmanager oder dergleichen lassen sich insbesondere ohne weiteres echtzeitfähige Systeme realisieren. Zur Echtzeitfähigkeit muss gewährleistet sein, dass auf eintreffende Daten beziehungsweise Interrupts, die insbesondere das Dateneintreffen signalisieren, innerhalb einer in keinem Fall

zu überschreitenden Maximalzeit reagiert werden kann. Dies kann beispielsweise geschehen durch einen Taskwechsel auf einen Interrupt hin und/oder, beispielsweise bei priorisierten Interrupts, durch Festlegung, dass ein gegebener Interrupt momentan zu ignorieren ist, wobei auch dies innerhalb einer bestimmten Zeit festzulegen ist. Ein Taskwechsel bei derartigen echtzeitfähigen Systemen wird typisch auf drei Arten erfolgen können, nämlich entweder dann, wenn ein Task eine bestimmte Zeit gelaufen ist (Watch-dog-Prinzip), bei Nichtzurverfügungstehen einer Ressource, sei es durch deren Blockade durch anderen Zugriff oder aufgrund von Latenzen beim Zugriff darauf, insbesondere in schreibender und/oder lesender Weise, das heißt bei Latenzen von Datenzugriffen und/oder beim Auftreten von Interrupts.

Mit der vorliegenden Erfindung kann die Echtzeitfähigkeit eines Datenverarbeitungslogikzellenfeldes nunmehr erreicht werden, indem eine oder mehrere von drei möglichen Varianten implementiert wird.

Eine erste Variante besteht darin, dass innerhalb einer von dem Scheduler beziehungsweise der CT ansprechbaren Ressource ein Wechsel zur Abarbeitung beispielsweise eines Interrupts erfolgt. Sofern die Ansprechzeiten auf Interrupts oder andere Anforderungen so groß sind, dass während dieser Zeit eine Konfiguration ohne Unterbrechung noch abgearbeitet werden kann, ist dies unkritisch, zumal während der Abarbeitung der aktuell laufenden Konfiguration auf jener Ressource, die für die Abarbeitung des Interrupts zu wechseln ist, eine Konfiguration zur Interruptabarbeitung vorgeladen werden kann. Die Auswahl der vorabzuladenden Interrupt-bearbeitenden Konfiguration ist z. B. durch die CT durchzuführen. Es ist möglich,

die Laufzeit der Konfiguration auf der für die Interruptbearbeitung freizugebenden bzw. zu wechselnden Ressource zu begrenzen. Verwiesen wird dazu auf PACT29/PCT(PCT/DE03/000942).

- 5 Bei Systemen, die schneller auf Interrupts reagieren müssen, kann es bevorzugt sein, eine einzelne Ressource, also beispielsweise eine separate XPP-Einheit und/oder Teile eines XPP-Feldes für eine solche Abarbeitung zu reservieren. Wenn dann ein schnell abzuarbeitender Interrupt auftritt, kann
10 entweder eine für besonders kritische Interrupts schon vorab vorgeladene Konfiguration abgearbeitet werden oder es wird sofort mit dem Laden einer Interrupt behandelnden Konfiguration in die reservierte Ressource begonnen. Eine Auswahl der jeweils für den entsprechenden Interrupt erforderlichen Konfiguration ist durch entsprechende Triggerung, Waveabarbeitung
15 usw. möglich.

- Es sei im Übrigen erwähnt, dass es mit den schon beschriebenen Methoden ohne weiteres möglich ist, eine instantane Reaktion auf einen Interrupt zu erhalten, indem über die Verwendung von LOAD/STORE-Konfigurationen eine Code-Reentranz erreicht wird. Hierbei wird nach jeder datenbearbeitenden Konfiguration oder zu gegebenen Zeiten, beispielsweise alle fünf oder zehn Konfigurationen eine STORE-Konfiguration ausgeführt
25 und dann eine LOAD-Konfiguration unter Zugriff auf jene Speicherbereiche ausgeführt, in die zuvor weggeschrieben wurde. Wenn sichergestellt wird, dass die von der STORE-Konfiguration benutzten Speicherbereiche so lange unberührt bleiben, bis durch Fortschreiten im Task eine weitere Konfiguration
30 sämtliche relevanten Informationen (Zustände, Daten) weggeschrieben hat, ist sichergestellt, dass bei Wiederladen, also Wiedereintritt in eine zuvor bereits begonnene, aber nicht zu

CONFIDENTIAL

4

Ende geführte Konfiguration oder Konfigurationskette wieder dieselben Bedingungen erhalten werden. Eine solche Zwischenschaltung von LOAD/STORE-Konfigurationen unter simultanem Schutz von noch nicht veralteten STORE-Speicherbereichen lässt sich automatisch ohne zusätzlichen Programmieraufwand sehr einfach generieren, z. B. von einem Compiler. Dort kann die Ressourcenreservierung gegebenenfalls vorteilhaft sein. Das bei der Ressourcenreservierung und/oder in anderen Fällen auf zumindest eine Menge hochpriorisierter Interrupts durch Vorabladen von bestimmten Konfigurationen reagiert werden kann, sei noch einmal erwähnt!

Eine weitere, besonders bevorzugte Variante der Reaktion auf Interrupts besteht dann, wenn zumindest eine der ansprechbaren Ressourcen eine Sequenziell-CPU ist, darin, auf dieser eine Interrupt-Routine abzuarbeiten, in welcher wiederum Code für das Datenverarbeitungslogikzellenfeld verboten ist. Mit anderen Worten wird eine Interrupt-Routine ausschließlich auf einer Sequenziell-CPU abgearbeitet, ohne dass XPP-Datenverarbeitungsschritte aufgerufen werden. Dies garantiert, dass der Verarbeitungsvorgang auf dem Datenverarbeitungslogikzellenfeld nicht zu unterbrechen ist und es kann dann eine Weiterabarbeitung auf diesem Datenverarbeitungslogikzellenfeld nach einem Taskswitch erfolgen. Obwohl damit die eigentliche Interrupt-Routine keinen XPP-Code besitzt, kann dennoch dafür gesorgt werden, dass auf einen Interrupt hin zu einem späteren, nicht mehr echtzeitrelevanten Zeitpunkt mit der XPP auf einen durch einen Interrupt und/oder eine Echtzeitanforderung erfassten Zustand und/oder Daten unter Verwendung des Datenverarbeitungslogikzellenfeldes reagiert werden kann.

L

5

10

15

Datenverarbeitungsvorrichtung mit einem Datenverarbeitungslogikzellenfeld und zumindest einer Sequenziell-CPU, dadurch gekennzeichnet, dass eine Ankopplung der Sequenziell-CPU und des Datenverarbeitungslogikzellenfeldes zum Datenaustausch in insbesondere blockweiser Form durch zu einem Cache-Speicher führende Leitungen möglich ist.

20.

25

2

5

10

15

Die Erfindung betrifft eine Datenverarbeitungsvorrichtung mit einem Datenverarbeitungslogikzellenfeld und zumindest einer Sequenziell-CPU. Hierbei ist vorgesehen, dass eine Ankopplung der Sequenziell-CPU und des Datenverarbeitungslogikzellenfeldes zum Datenaustausch in insbesondere blockweiser Form durch zu einem Cache-Speicher führende Leitungen möglich ist.

20

25

.....

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☒ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.